AIUTARE: A Modular Benchmarking Framework

Finnbarr O'Callahan, Federico Mora, David Tang, Professor Marsha Chechik

0) Introduction

- Testing programs on thousands of benchmark cases produces a substantial amount of data.
- AIUTARE abstracts the process of running programs and organizes output in easy-to-manipulate mongoDB databases.

1) Case Study: SMT Solution Cross-Checking

- SMT solvers can be used to test web security and verify safety-critical software.
- SMT solvers are given SMT queries...

variables { String x, String y, String m, String n } { y + x = n + m, n = '', len(x) = int(m) } constraints

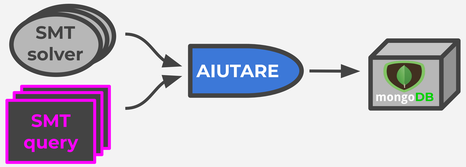
- ...and return *SAT* or *UNSAT* if variables can be assigned values to satisfy constraints.
- If *SAT*, solver also returns a solution...

String x = '', String y = '0', String m = '0', String n = '' | y + x = n + m, n = "", len(x) = int(m)

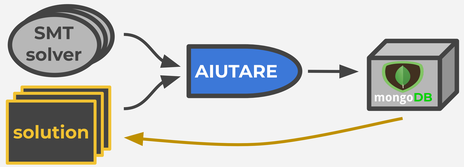
- ...with concrete values, ***which can be wrong!***

2) Methods

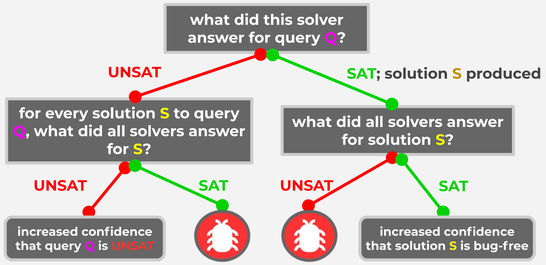
- 3 SMT solvers vs. 18,000 SMT queries.
- Run solvers on queries using AIUTARE:



- For every time a solver returns *SAT*, store the solution that is given as proof.
- Run all the solvers again, this time on all the solutions that have been generated:



- Finally, go through all results now in the mongoDB database to find 2 types of bugs:



3) Results

- An example bug our method caught, bugged solution vs. correct solution:

solver: Z3	solver: CVC4
String x = ""	String x = ""
String y = ""	String y = "0"
String m = ""	String m = "0"
String n = ""	String n = ""
y + x = n + m ✓	y + x = n + m ✓
n = "" ✓	n = "" ✓
len(x) = int(m) X	len(x) = int(m) ✓
3/3 UNSAT	3/3 SAT

- Overall testing results:
  - 5 solutions were found to have bugs.
  - 4 faulty answers of "UNSAT" were disproven by counterexample.

4) Conclusions

- Even the best SMT solvers still have bugs!
- Solvers which are used for safety-critical applications should continue to be tested and cross-checked on a wide variety of benchmark cases as we have done here.
- AIUTARE is easy to use and extend.
- All SMT solution cross-checking implemented in < 200 lines of code!

5) Next Steps

- For any programs that take input as command line parameters and output either true or false as seen below:

```
$ programA query1.txt
$ programA query2.txt
$ programA query3.txt
$ programB query1.txt
$ programB query2.txt
$ programB query3.txt
```

→ SAT / TRUE / YES or UNSAT / FALSE / NO

- AIUTARE can efficiently run any of these programs in parallel and organize all output data in MongoDB.
- We plan to apply this tool for:
  - Performance/runtime analysis of SMT and SAT solvers.
  - Testing equivalence checkers, such as CLEVER (under development at UofT).
  - Generating plots and regression models.

6) Acknowledgements & Contact

- Thank you to Professor Chechik and the whole UofT Modeling research group, as well as to NSERC for funding.

- *Use AIUTARE on your program too!* github.com/FinnbarrOC/aiutare